Обработка исключений

|  |  |  |
| --- | --- | --- |
| **№ урока:** | 23 | **Курс:** |

C# Essential

|  |
| --- |
| **Средства обучения:** |

Компьютер с установленной Visual Studio

# Обзор, цель и назначение урока

Рассмотрение обработки исключений.

**Изучив материал данного занятия, учащийся сможет:**

* Понимать и использовать техники обработки исключительных ситуаций.
* Понимать работу класса Exception и конструкций try - catch - finally.

# Содержание урока

1. Обработка исключений.
2. Ключевое слово try.
3. Ключевое слово catch.
4. Ключевое слово finally.
5. Ключевое слово throw.
6. Класс Exception.

# Резюме

* Обработка исключительных ситуаций (exception handling) — механизм, предназначенный для описания реакции программы на ошибки времени выполнения и другие возможные проблемы

(исключения), которые могут возникнуть при выполнении программы и приводят к невозможности (бессмысленности) дальнейшей отработки программой её базового алгоритма.

* В С# исключения представлены в виде классов. Все классы исключений должны быть производными от встроенного в С# класса Exception, являющегося частью пространства имен System. Следовательно, все исключения в C# являются подклассами класса Exception.
* Главное преимущество обработки исключительных ситуаций заключается в том, что она позволяет автоматизировать получение большей части кода, который раньше приходилось вводить в любую крупную программу вручную для обработки ошибок. Так, если программа написана на языке программирования без обработки исключительных ситуаций, то при неудачном выполнении методов приходится возвращать коды ошибок, которые необходимо проверять вручную при каждом вызове метода. Это не только трудоемкий, но и чреватый ошибками процесс.
* Основу обработки исключительных ситуаций в С# составляет пара ключевых слов try и catch.

Эти ключевые слова действуют совместно и не могут быть использованы порознь.

* Когда исключение генерируется оператором try, оно перехватывается составляющим ему пару оператором catch, который затем обрабатывает это исключение. В зависимости от типа исключения выполняется и соответствующий оператор catch. Так, если типы генерируемого исключения и того, что указывается в операторе catch, совпадают, то выполняется именно этот оператор, а все остальные пропускаются.
* Если исключение не генерируется, то блок оператора try завершается как обычно, и все его операторы catch пропускаются. Выполнение программы возобновляется с первого оператора, следующего после завершающего оператора catch. Таким образом, оператор catch выполняется лишь в том случае, если генерируется исключение.
* Операторы catch выполняются по порядку их следования в программе. Но при этом выполняется только один блок catch, в котором тип исключения совпадает с типом генерируемого исключения. А все остальные блоки catch пропускаются.

# Закрепление материала

* Что такое исключение?
* Что такое конструкция try - catch?
* Что такое конструкция try - catch - finally?
* Как создать пользовательское исключение?
* В каких случаях не срабатывает блок finally?